

Classifying Documents using Ruby

Paul Dix

paul@pauldix.net

<http://www.pauldix.net>

Roadmap

- Introduction to Classifiers
- The Document Vector Model
- Feature Selection
- Overview of Classification Algorithms
- Testing
- Closing Observations

What are classifiers?

Supervised Machine Learning

Useful For

- Language Identification
- News Category
- Blog Category
- Spam Detection
- Sentiment Detection

Things You Do

- Get Training Data
- Document Preprocessing
- Feature Selection (optional)
- Train the Classifier
- Test and Update

Roadmap

- Introduction to Classifiers
- **The Document Vector Model**
- Feature Selection
- Overview of Classification Algorithms
- Testing
- Closing Observations

In Text Classification We
Represent Documents
as a Vector of Features.

Basic Representation

- clean text of punctuation and numbers
- lowercase everything
- split on spaces
- stem all words
- return an array of terms and counts



show document.rb

Document by Term (or feature) Matrix

show output.train

Terms

Docs

	ruby	the	python	erlang
1	2	17	0	0
2	1	20	1	0
3	0	12	0	1

Additional Features

- You can create complicated preprocessing
- Include link structures
- Include metadata
- Include social data

High Dimensionality

- Noisy features
- Non-discriminating features
- Words that are too similar

Stemming

- Porter Stemmer - Dr. Martin Porter '79
- "cats".stem # => "cat"
- "international".stem # => "intern"
- "finalize".stem # => "final"
- "ruby".stem # => "rubi"

Roadmap

- Introduction to Classifiers
- The Document Vector Model
- **Feature Selection**
- Overview of Classification Algorithms
- Testing
- Closing Observations

Advantages

- Limits Size of Vocabulary
- Increases Classification Accuracy
- Avoids Noisy Features and Overfitting

Different Methods

- Mutual Information
- Chi-squared
- Frequency Based


The Process

- add each document from training set
- calculate best features
- select top features



Feature Extraction

- Selected Features from Each Document
- Consistent Representation to Classifier
- Avoiding A Sparse Document-Feature Matrix



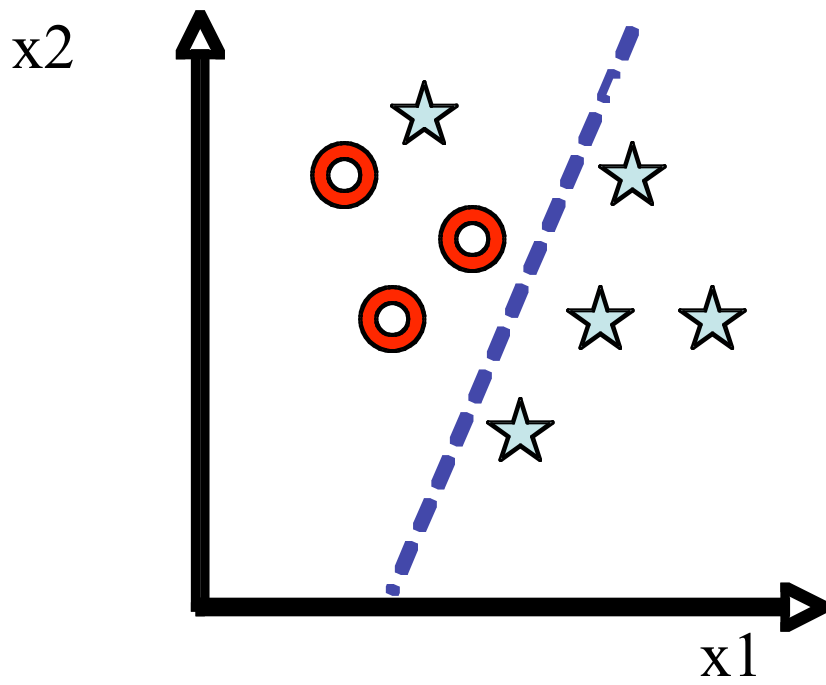
```
show  
feature_extractor.rb
```

Roadmap

- Introduction to Classifiers
- The Document Vector Model
- Feature Selection
- **Overview of Classification Algorithms**
- Testing
- Closing Observations

Contiguity Hypothesis

Documents in the same class form a contiguous region
and regions of different classes don't overlap.



☆ topic2
⊙ topic1

Linear Classifiers

- Finds a Hyperplane
- Simple decision boundary

Naive Bayes



- Classifier gem
- Generative Model using Probabilities
- Bayes Formula

$$P(d \in C | F_1, F_2, \dots, F_k) = \frac{P(F_1, F_2, \dots, F_k | d \in C)P(d \in C)}{P(F_1, F_2, \dots, F_k)}$$

Why Naive

- Positional Independence
- Conditional Independence
- Also known as Idiot's Bayes

NB Advantages

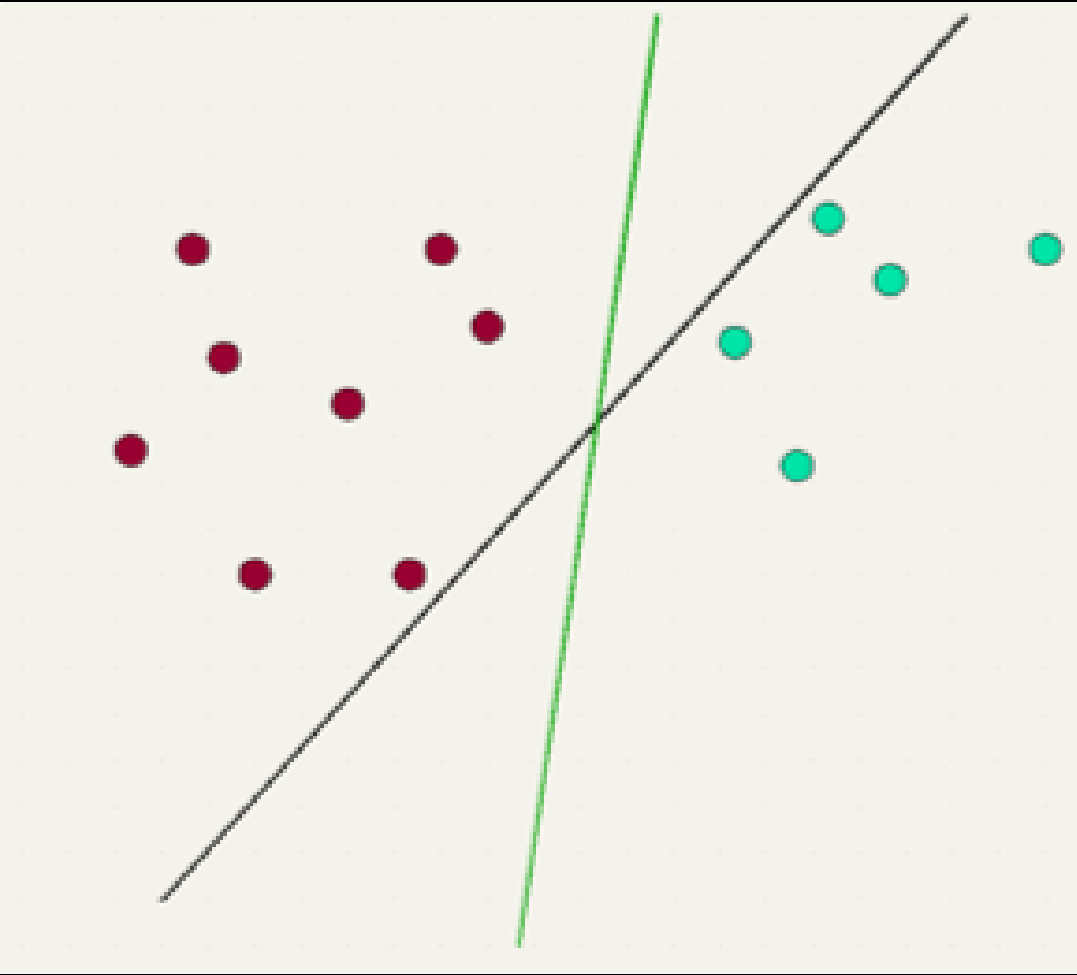
- Simple
- Fast
- Effective for Text Classification

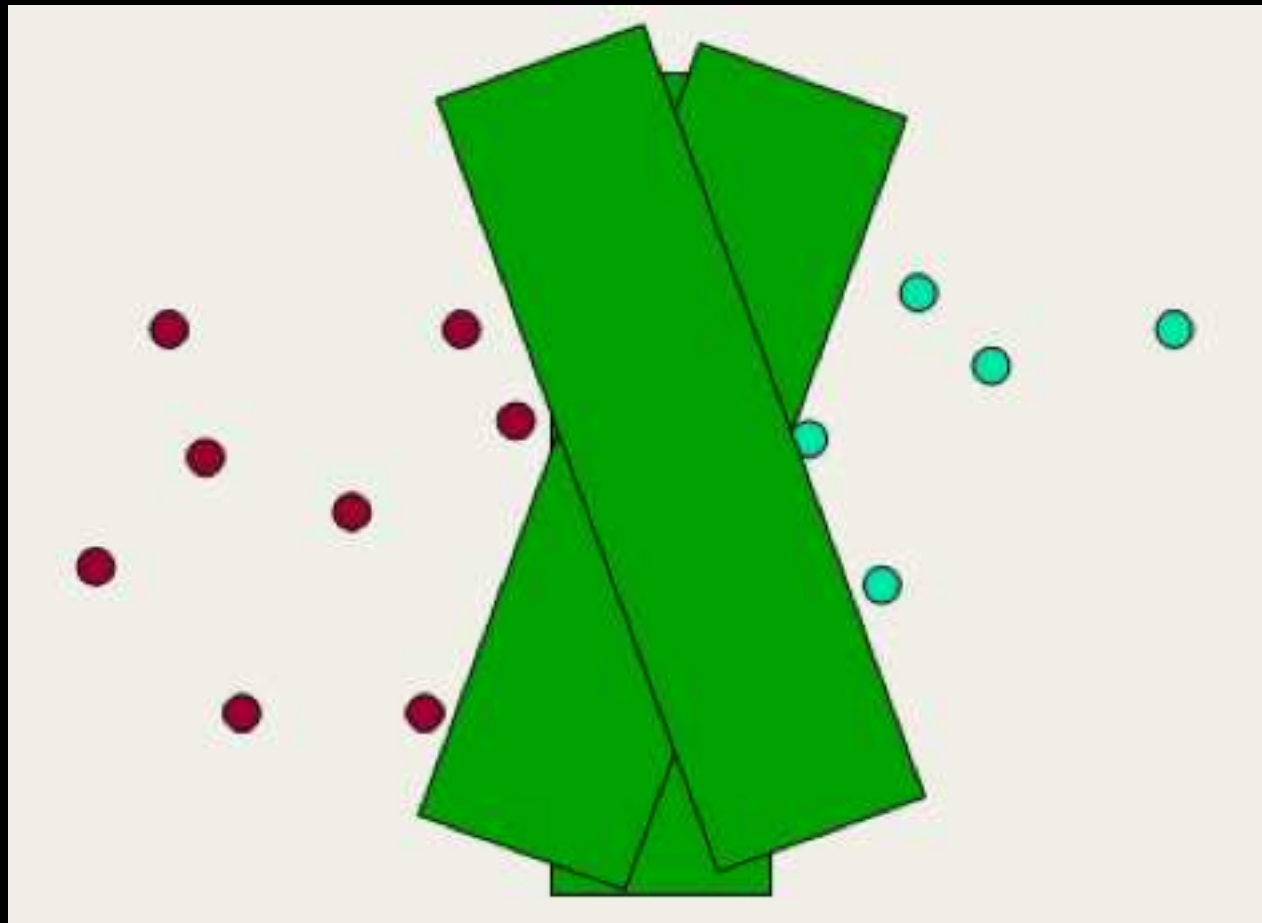
Other Considerations

- Binomial vs. Multinomial Models
- Probabilities of Classes Based on Training Set

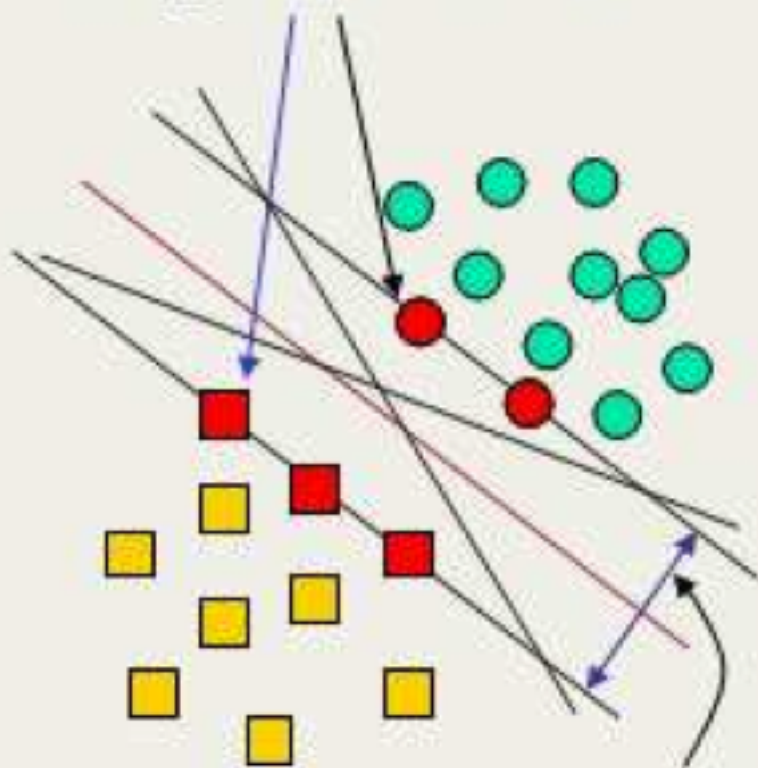
Support Vector Machines

- Vector Based Linear Classifier
- Decision Boundary Maximally far Away
- Ruby SVM





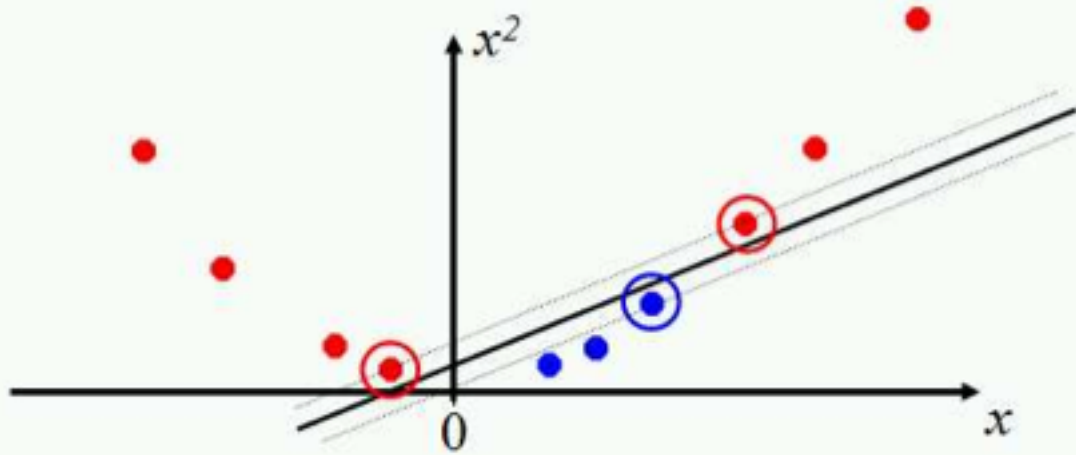
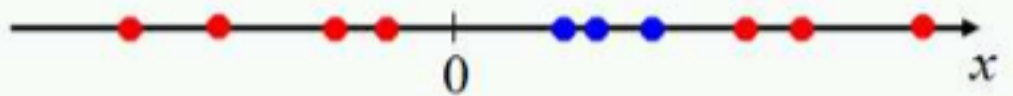
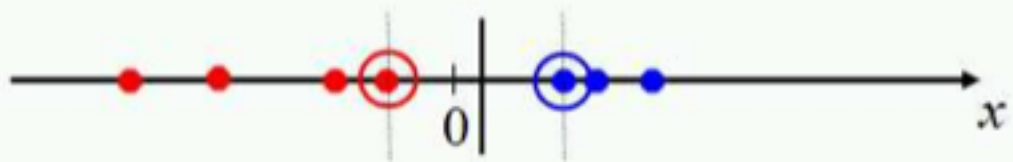
Support vectors



Maximize
margin

Kernels

- aka “the kernel trick”
- a function to transform the feature space
- non-linear classification
- linear, polynomial, radial basis function, sigmoid



Perceptron

- Neural Network
- Linear Classifier
- Converges through iterations
- Not guaranteed to converge

k Nearest Neighbors

- non-linear
- arbitrarily complex decision boundary
- no traditional training
- can be slow for large training set

Others

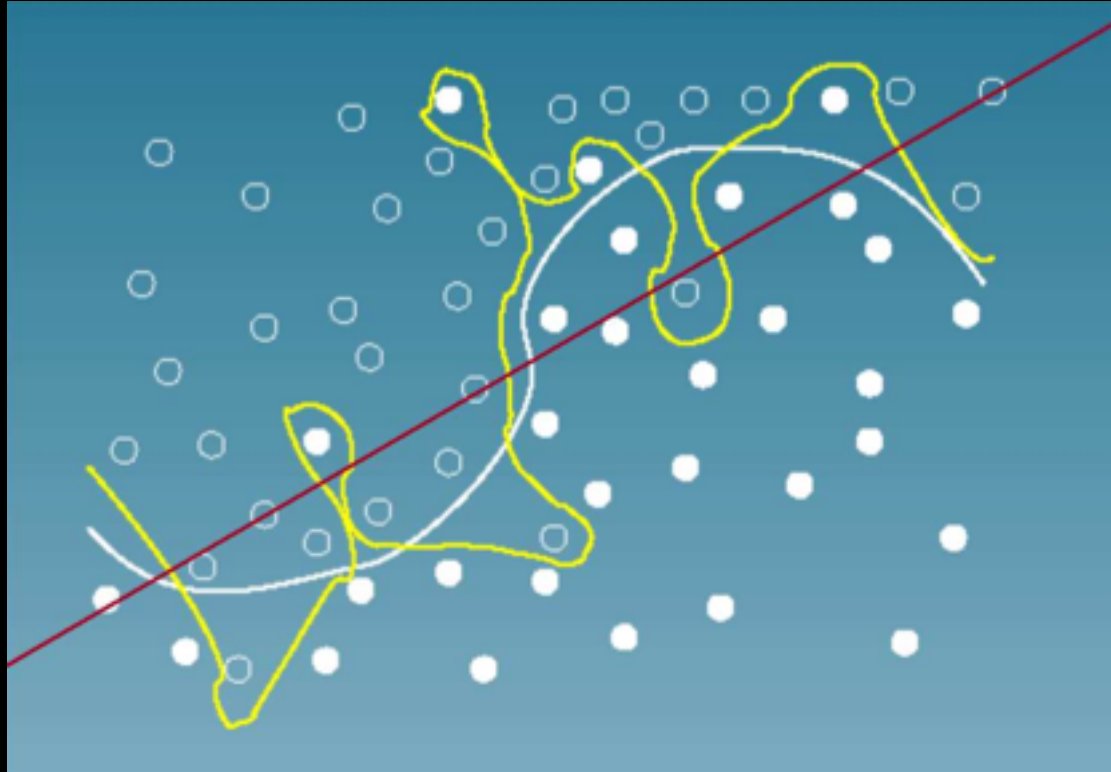
- Decision Trees
- Rocchio
- Compression
 - Squish by Bob Aman
- Latent Semantic Analysis
- Weka

Multiple Classes

- Any of classification
- One of classification

Bias-variance Tradeoff

- Linear classifiers are high bias
- Non-linear classifiers are high variance



Roadmap

- Introduction to Classifiers
- The Document Vector Model
- Feature Selection
- Overview of Classification Algorithms
- **Testing**
- Closing Observations

- Each classification task different
- Constant tuning

Basic Methods

- Training and Test sets
- Accuracy - $\frac{\# \text{ correct}}{\# \text{ total}}$
- Cross Validation
 - 10 fold is common

Closing Observations

- Some easier than others
 - Can expect $> 90\%$ on easy tasks
- Best for triage
- Naive Bayes simplest to work with